Oxford Cambridge and RSA

Wednesday 08 Feb 2024 – Afternoon

A Level Computer Science

H446/02 Algorithms and programming (part 2)

Time allowed: 70 minutes

You can use:
- a ruler (cm/mm)
- an HB pencil

Do not use:
- a calculator

Please write clearly in black ink. **Do not write in the barcodes.**

Centre number | | | | | | Candidate number | | | | |

First name(s) ____________________

Last name ____________________

INSTRUCTIONS
- Use black ink.
- Write your answer to each question in the space provided. If you need extra space use the lined pages at the end of this booklet. The question numbers must be clearly shown.
- Answer **all** the questions.

INFORMATION
- The total mark for this paper is **140**. (part1 71 marks and part 2 69 marks
- The marks for each question are shown in brackets **[ ]**.
- Quality of extended response will be assessed in questions marked with an asterisk (*).

ADVICE
- Read each question carefully before you start your answer.

Part 1 - / 71
Part 2 - / 69
Total - /140 Grade:

Turn over

2 The pseudocode function `binarySearch()` performs a binary search on the array `dataArray` that is passed as a parameter. The function returns the array index of `searchValue` within the array, and `-1` if it is not in the array.

(a) The pseudocode binary search algorithm is incomplete.

(i) Complete the algorithm by filling in the missing statements.

```
function binarySearch(dataArray:byref, upperbound, lowerbound, ............ )
  while true
    middle = lowerbound + ((upperbound – lowerbound) ............ )
    if upperbound < lowerbound then

      return ..................................................
    else
      if dataArray[middle] < searchValue then

        lowerbound = ..............................................................................................................................
      elseif dataArray[middle] > searchValue then

        upperbound = ..............................................................................................................................
      else
        return ..........................................................................................................
      endif
    endif
  endwhile
endfunction
```

[6]

(ii) The algorithm uses a while loop.

State a different type of loop that could be used instead of the while loop in the given algorithm.

....

. ... **[1]**

(b) The tables below show possible Big O complexities for the worst-case space, best-case space and average time for search algorithms.

Tick the worst-case space complexity for a binary and linear search.

	Binary search	**Linear search**
O(log(n))		
O(1)		
O(n)		

Tick the best-case space complexity for a binary and linear search.

	Binary search	**Linear search**
O(log(n))		
O(1)		
O(n)		

Tick the average time complexity for a binary and linear search.

	Binary search	**Linear search**
O(log(n))		
O(1)		
O(n)		

[6]

(c) Identify **one** situation where a linear search is more appropriate than a binary search.

....

. ... **[1]**

4* Anna currently writes her program code in a text editor and then runs the compiler.

She has been told that using an Integrated Development Environment (IDE) would be more helpful.

Discuss the benefits of Anna using an IDE to write and test her program rather than using a text editor.

[9]

6 Amy's processor makes use of pipelining during the fetch-decode-execute cycle.

(a) The processor's pipeline consists of the following stages:

- Fetching the instruction from memory
- Decoding the instruction
- Executing the instruction.

Instructions A, B, C and D need to be processed.

Identify the stage(s) and instruction(s) run during each pipeline below.

Pipeline 1 ....

....

....

Pipeline 2 ....

....

....

Pipeline 3 ....

....

....

Pipeline 4 ....

....

....

[4]

(b) Explain why pipelining can improve the performance of the processor.

....

....

....

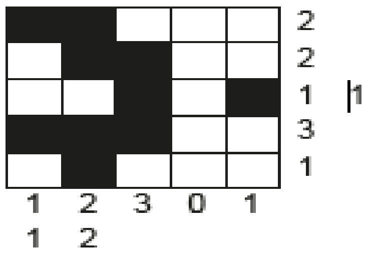
. ... **[2]**

Section B

Answer **all** the questions.

8 A Nonogram is a logic puzzle where a player needs to colour in boxes. The puzzle is laid out as a grid and each square needs to be either coloured black or left white.

The numbers at the side of each row and column tells the player how many of the boxes are coloured in consecutively. Where a row has two or more numbers, there must be a white square between the coloured squares.



In this example:

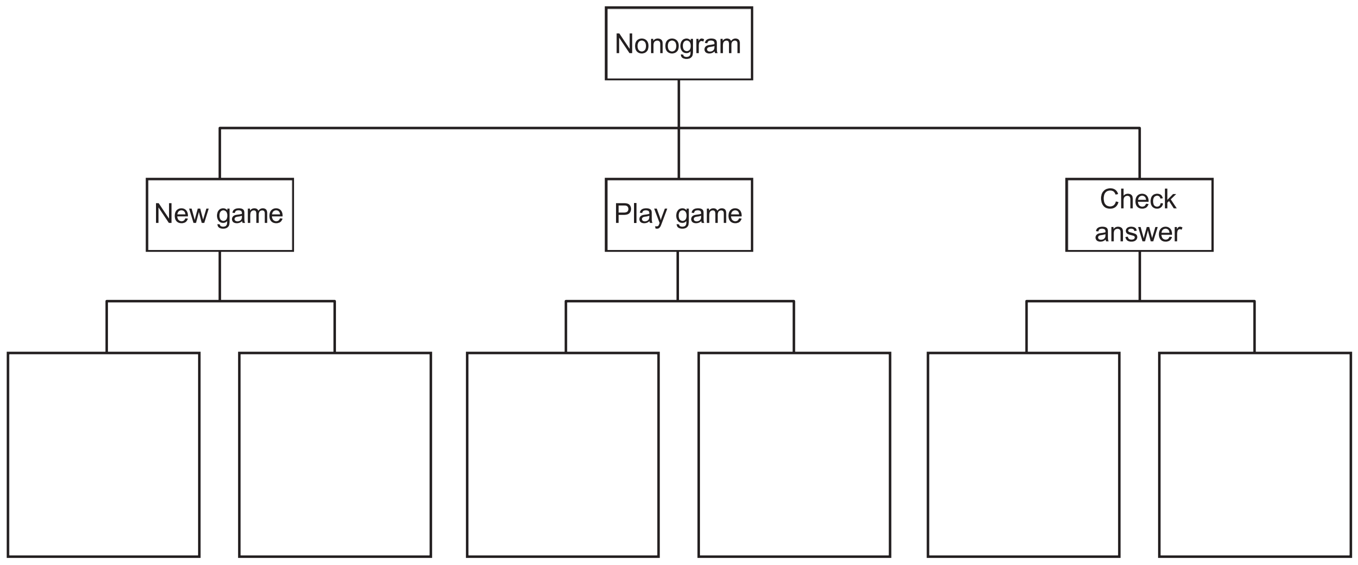
- the first column has 1 1, this means there must be two single coloured boxes in this column. There must be at least 1 white box between them.
- the first row has 2, this means there must be two consecutively coloured boxes in the row.

Juan is creating a program that will store a series of Nonograms for a user to play. The game will randomly select a puzzle and display the blank grid with the numbers for each row and column to the user.

The user plays the game by selecting a box to change its colour. If the box is white it will change to black and if it is black it will change to white. The user can choose to check the answer at any point, and the game will compare the grid to the answers and tell the user if they have got it correct or not.

(a) Juan is creating a structure diagram to design the game.

(i) Complete the structure diagram by adding another layer for New game, Play game and Check answer.



[3]

(ii) A structure diagram is one method of showing the decomposition of a problem.

Explain why decomposing a problem can help a developer design a solution.

....

....

....

. ... **[2]**

(iii) Identify **one** input, **one** process and **one** output required for the game.

Input ....

Process ....

Output ....

[3]

(b) Juan uses the structure diagram to create a modular program with a number of subroutines. The program will use two integer 2-dimensional arrays to store the puzzles:

- `puzzle(5,5)` stores the solution
- `answerGrid(5,5)` stores the user's current grid.

A `0` represents a white box and a `1` represents a black box.

(i) Juan creates a function, `countRow()`, to count the number of coloured boxes in one row and return the number of consecutive coloured boxes in that row. If there is more than one set of coloured boxes in the row, these are joined together and the string is returned.

For example, in the following grid `countRow` for row 0 will return `"2"` as a string, and `countRow` for row 2 will return `"1 1"` as a string. If there are no 1s in a row, then `"0"` is returned as a string.

1	1	0	0	0
0	1	1	0	0
0	0	1	0	1
1	1	1	0	0
0	1	0	0	0

Complete the pseudocode algorithm `countRow()`.

```
function countRow(puzzle:byref, rowNum:byval)
  count = 0
  output = " "
  for i = 0 to ………………………………………
    if puzzle[rowNum, i] ==...................then
      count = count + 1
    elseif count >= 1 then
      output = output + str(………………………………………) + " "
      count = 0
    endif
  next i
  if count>= 1 then
      output=output+str(count)
  elseif output == "" then
      output = "…................ "
  endif
  return ………………………………………
endfunction
```

[5]

(ii) Explain the purpose of line 03 in the function `countRow`.

....

....

....

. ... **[2]**

(iii) Describe the purpose of branching and iteration in the function `countRow`.

....

....

....

....

....

. ... **[3]**

(iv) The procedure `displayRowAnswer()` takes `puzzle` as a parameter and outputs the value in each box. Each box in a row is separated by a space. At the end of each row there are two spaces and (by calling the function `countRow` from **part 8(b)(i)**) the clue values for that row.

For example the puzzle below:

1	1	0	0	0
0	1	1	0	0
0	0	1	0	1
1	1	1	0	0
0	1	0	0	0

Would output:

```
1 1 0 0 0   2
0 1 1 0 0   2
0 0 1 0 1   1 1
1 1 1 0 0   3
0 1 0 0 0   1
```

Write pseudocode or program code for the procedure `displayRowAnswer()`.

....

....

....

....

....

....

....

....

....

....

....

.... **[6]**

(v) The function `checkWon()` takes `answerGrid` and `puzzle` as parameters and compares each element in the grids. If they are identical, it returns `true`, otherwise returns `false`.

```
01 function checkWon(puzzle)
02   for row = 0 to 4
03     for column = 0 to 4
04       if puzzle[row, column] == answerGrid[row, column] then
05       return false
06       endif
07     next column
08   next column
09   return true
10 endfunction
```

There are **three** logic errors in the function `checkWon`.

State the line number of each error and give the corrected line.

Error 1 line number ....

Error 1 correction ....

Error 2 line number ....

Error 2 correction ....

Error 3 line number ....

Error 3 correction ....

[3]

(c)* Juan passed the two arrays as parameters, but he did consider making them globally accessible.

Compare the use of global and local variables and data structures in this program. Include the use of parameters and program efficiency in your answer.

[9]

(d) Juan wants to create a program that will generate new Nonograms with different grid sizes. For example a Nonogram with a 10 × 10 grid or a 5 × 20 grid.

Describe how the program could be written to automatically generate a new Nonogram.

....

....

....

....

....

....

....

. ... **[4]**

END OF QUESTION PAPER